UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/783,598	02/20/2004	David Wortendyke	MS1-2021US	1564

22801 7590 10/06/2008

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER	
WU, YICUN	

ART UNIT	PAPER NUMBER
2169	

MAIL DATE	DELIVERY MODE
10/06/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/783,598	WORTENDYKE ET AL.	
	Examiner	Art Unit	
	YICUN WU	2169	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 September 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,7,8,10-15,19-21,25-28,30,31,33,35-37 and 40-43* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,7,8,10-15,19-21,25-28,30,31,33,35-37 and 40-43* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

III. DETAILED ACTION

1. Claims 1-5,7,8,10-15,19-21,25-28,30,31,33,35-37 and 40-43 are presented for examination.

Remarks

2. Specification:

The specification, on page 4, filed on 9/25/2008 introduced new matter.

Claims: Please update claims 40-42 "computer readable media" for lack of antecedent basis.

Rejection:

New ground of rejection has been applied.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matt~ any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 1-5, 7-8 31, 33 and 35-36 are rejected under 35 U.S.C. 101 because the claimed invention is directed non-statutory subject matter.

Claims 1-5, 7-8 31, 33 and 35-36 lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 USC 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a

composition of matter. As such, they fail to fall within a statutory category. They are, at best, functional descriptive material *per se*.

Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." Both types of "descriptive material" are nonstatutory when claimed as descriptive material *per se*, 33 F.3d at 1360, 31 USPQ2d at 1759. When functional descriptive material is recorded on some computer-readable medium, it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994)

Merely claiming nonfunctional descriptive material, i.e., abstract ideas, stored on a computer-readable medium, in a computer, or on an electromagnetic carrier signal, does not make it statutory. See *Diehr*, 450 U.S. at 185-86, 209 USPQ at 8 (noting that the claims for an algorithm in *Benson* were unpatentable as abstract ideas because "[t]he sole practical application of the algorithm was in connection with the programming of a general purpose computer.").

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which the subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-4, 7-8, 10-13, 15, 37 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Aguilera et al. (NPL #1, Matching Events in a Content- based Subscription System), in view of Graefe et al. (NPL #2, Dynamic Query Evaluation Plans) further in view of Syeda-Mahmood et al. (U.S. Patent No. 6,757,686).

As per claims 1,10 and 37, Aguilera et al. discloses a method for updating a filter engine opcode tree, comprising the following steps:

(a) compiling a new query to derive a series of opcode objects (Sec. 3, Para. 2, "each subscription is a conjunction of elementary predicates);

(b) traversing the opcode tree according to the series of opcode objects until an opcode object is encountered that is not included in the opcode tree, opcode objects being represented in the opcode tree as opcode nodes (App. A, Para. 2, lines 5-7);

(c) adding new opcode tree opcode nodes to correspond to the encountered opcode object and subsequent opcode objects in the series of opcode objects (App. A, Para. 2, lines 7-8).

the optimized lookup procedure comprises an interval tree function to optimize numeric interval queries; and

(f) restoring the optimized branch node to a generic branch node when the optimized branch node is no longer more efficient than the generic branch code.

However, Aguileraet al. does not explicitly disclose (d) updating a branch node in the opcode tree to add a reference to the new opcode nodes, the branch node being referenced from a parent opcode node that corresponds to a last opcode object from the series of opcode objects

that was found in the traversal of the opcode tree; and (e) implementing an optimized branch node that includes an optimized indexed lookup procedure, wherein the indexed lookup procedure is configured to return a set of (key, value) pairs, a single (key, value) pair corresponding to one branch.

However, Graefe et al. discloses updating a branch node in the opcode tree to add a reference to the new opcode nodes, the branch node being referenced from a parent opcode node that corresponds to a last opcode object from the series of opcode objects that was found in the traversal of the opcode tree (Pg. 362, second full paragraph, "this operator provides the same open, next, close protocol as the other operators and can therefore be inserted into a query plan at any location." This, also, shows that the 'choose plan operator' would be its own opcode node, when used in conjunction with Aguilera et al. Additionally, pg. 361, last paragraph of section 5, "If there is no gain in using a dynamic access module, the decision tree can by an empty function"); (e) implementing an optimized branch node that includes an optimized indexed lookup procedure, wherein the indexed lookup procedure is configured to return a set of (key, value) pairs, a single (key, value) pair corresponding to one branch. (This limitation is nothing more than a recitation of the steps taken by a hash algorithm, or related types of indexing methods, Applicant's spec. pg. 22, lines 3-7.

As such, it would have been obvious to one of ordinary skill in the art at the time of the invention to use an efficient indexing method, when appropriate. For instance, Graefe et al. page 360 "using a hash index"). It would have been obvious to one of ordinary skill in the art at the time of the invention to evaluation system of Aguilera et al. with the choose plan operators of Graefe et al. in order to increase its efficiency as described in Graefe et al.

Syeda-Mahmood teaches an interval tree function (abstract).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Aguileraet al to include an interval tree function.

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Aguileraet al by the teaching of Syeda-Mahmood to include an interval tree function with the motivation to provide an efficient data structure and algorithm for the solution as taught by Syeda-Mahmood (col. 1, lines 63-67).

As per claim 2, Aguilera et al. as modified, also discloses wherein one or more of the steps are performed dynamically at runtime (Aguilera et al., pg. 2, Col. 1, lines 22-24). It is a well-known goal of subscription server systems, as disclosed by Aguilera et al., to maximize availability. As such, it is anticipated by Aguilera et al. that the incremental updates would be done at run-time.

As per claim 3, Aguilera et al. as modified, also discloses further comprising performing steps (b) and (c) in a component of the filter engine (Aguilera et al., code listing, pg. 9, the pre-processing steps are a procedure contained within the tree matching algorithm, and as such, do not pertain to a stand alone program, and are therefore inherently part of the tree matching algorithm (filter engine)).

As per claim 4, Aguilera et al. as modified, also discloses further comprising executing the opcode tree against an input to evaluate the new query and one or more other queries against the input (Aguilera et al., Sec. 2, Par. 2, lines 1-7).

As per claim 7, Aguilera et al. as modified, discloses the branch node further comprising updating the branch node to include an indexed lookup routine that references several dependent opcode nodes that perform a similar function (Aguilera et al., Section 5, dynamic query evaluation plans include a decision procedure which is used at initialization time, and intermittently thereafter to choose the optimal evaluation strategy).

As per claim 8, Aguilera et al. as modified, discloses further comprising analyzing opcode nodes that depend from the branch node and including the indexed lookup routine only if including the indexed lookup routine provides more efficient processing of the dependent nodes that a generic branch node processing routine (Graefe et al. Section 5, dynamic query evaluation plans include a decision procedure which is used at initialization time, and intermittently thereafter to choose the optimal evaluation strategy).

As per claim 11, Aguilera et al. as modified, discloses the opcode merger further configured to traverse the opcode tree to determine if an opcode node corresponding to the new query already exists in the opcode tree and add new opcode nodes that correspond to query opcode blocks that are not already included in the opcode tree (Aguilera et al., App. A, pre-processing).

As per claim 12, Aguilera et al. as modified, also discloses wherein opcode nodes corresponding to opcode blocks included in a common prefix are represented as a shared segment in the opcode tree (Aguilera et al., App. A).

As per claim 13, Aguilera et al. as modified, also discloses wherein queries are merged into the opcode tree dynamically at runtime (Aguilera et al., pg. 2, Col. 1, lines 22-24). It is a well-known goal of subscription server systems, as disclosed by Aguilera et al., to maximize availability. As such, it is anticipated by Aguilera et al. that the incremental updates would be done at run-time.

As per claim 14, although Aguilera et al. as modified, does not explicitly disclose further comprising Xpath queries in the plurality of queries, it would be obvious to one of ordinary skill in the art to use this system in conjunction with Xpath queries. Evidence for this can be seen by the fact that Aguilera et al. has been cited multiple times by papers dealing with XML and Xpath, and listed as related research. The systems are not incompatible, as Aguilera et al.'s tests could be tests against Xpath attributes.

As per claim 15, Aguilera et al. as modified, discloses the compiler being further configured to create opcode objects that are configured to merge themselves into an appropriate location in the opcode tree (Aguilera et al. App A, pre-processing algorithm).

As per claim 17, Aguilera et al. as modified, does not explicitly disclose the indexed lookup routine further comprising one of the following routines: a hash routine; a routine that uses tries; an interval tree routine. However, Graefe et al. discloses the indexed lookup routine further comprising one of the following routines: a hash routine; a routine that uses tries; an interval tree routine (Graefe et al. pg. 360). It would obvious to one of ordinary skill in the art at the time of the invention to further modify the Aguilera et al. combination to utilize an efficient indexing method based on the attribute in question, as taught by Graefe et al. As per claim 18, Aguilera et al. as modified, implicitly has the property that the opcode merger is further configured to restore an optimized branch node to a generic branch node when the optimized branch node is no longer more efficient that the generic branch node (last paragraph of section 5, lines 4-6). The choose-plan is evaluated each time, and causes the most efficient query evaluation technique to be used. If this is the usual sequential method, then it does so.

As per claim 41, this claim is substantially the same as claim 10 above, and is rejected on the same grounds.

8. Claims 5, 19-21, 23-28, 30, 35-36, 40, 42-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Aguilera et al. (NPL #1, Matching Events in a Content-based Subscription System), in view of Graefe et al. (NPL #2, Dynamic Query Evaluation Plans) as applied to claims 1-4, 7-13, 15, 17, 37, 39, 41 above, and in further view of Java API (NPL #3, excerpt from Java 2 Platform Std. Ed. V1.4.2) and further in view of Syeda-Mahmood et al. (U.S. Patent No. 6,757,686).

As per claim 5 Aguilera et al. as modified, does not explicitly disclose receiving a request to remove a first query from the opcode tree; identifying one or more opcode nodes in the opcode tree that correspond to the first query; removing any identified opcode node that does not correspond to a second query. However, the Java API discloses methods for both inserting and removing nodes (Java API: method summary). If utilized by Aguilera et al. as modified, the system would implicitly perform the steps of: receiving a request to remove a first query from the opcode tree;

identifying one or more opcode nodes in the opcode tree that correspond to the first query; removing any identified opcode node that does not correspond to a second query (Aguilera et al. App. A, Java API, method summary).

It would have been obvious to one of ordinary skill in the art at the time of the invention to provide a method of removal similar to the method of insertion as taught by the Java API to the Aguilera et al. combination in order to maintain the disclosed space complexity. It is disclosed that is linear with respect to the number of subscriptions (Sec. 4, heading "Space Complexity").

As per claim 20, Aguilera et al. as modified, teaches opcode objects that are further configured to merge into the opcode tree only if an identical opcode object corresponding to a similar query prefix is not already included in the opcode tree (Aguilera et al. App. A). This functionality is implicit if the nodes as provided by claim 19 are used with the matching tree of

Aguilera et al. The system would not provide the increased efficiency intended by Aguilera et al. if it allowed duplication of nodes.

As per claim 21, Aguilera et al. as modfiied, teaches it would have been obvious to one of ordinary skill in the art to use the Xpath language in the implementation of this system as Xpath was considered to be the standard at the time of the invention Important factors when deciding implementation details, such as language, are interoperability and future usage Using a language and specification that are standard increases the interoperability and lifespan of programs (See IEEE: Benefits of Standards, NPL #7).

As per claim 23, Aguilera et al. as modified, does not explicitly disclose an optimized branching function further comprises a function selected from the following list of functions: a hash function, an interval tree function, and a function utilizing tries.

However, Graefe et al. discloses an optimized branching function further comprises a function selected from the following list of functions: a hash function, an interval tree function, and a function utilizing tries (Graefe et al. pg. 360).

It would obvious to one of ordinary skill in the art at the time of the invention to further modify the Aguilera et al. combination to utilize an efficient indexing method based on the attribute in question, as taught by Graefe et al.

As per claim 24, Aguilera et al. as modified, implicitly has the property wherein the branch node is configured to recognize a context where the optimized branching function is no

longer efficient and to resort to its previous function if such a context develops (last paragraph of section 5, lines 4-6). The choose-plan is evaluated each time, and causes the most efficient query evaluation technique to be used. If this is the usual sequential method, then it does so.

As per claim 25, Aguilera et al. as modified, discloses the compiler is configured to receive the query and generate the opcode at runtime (Aguilera et al. pg. 54, Col. 1, lines 22-24); and the opcode node is configured to merge itself into the opcode tree at runtime (pg. 54, Col. 1, lines 22-24).

It is a well-known goal of subscription server systems, as disclosed by Aguilera et al., to maximize availability. As such, it is anticipated by Aguilera et al. that the incremental updates would be done at run-time. In addition, moving portions of this functionality to the opcode nodes does not make the system unable to stay active while updating.

As per claim 27, Aguilera et al. as modified, discloses perform[ing] the recited steps dynamically at runtime (pg. 2, Col. 1, lines 22-24). It is a well-known goal of subscription server systems, as disclosed by Aguilera et al., to maximize availability. As such, it is obvious in view of the teachings of Aguilera et al. that the incremental updates would be done at run-time. In addition, moving portions of this functionality to the opcode nodes does not make the system unable to stay active while updating.

As per claim 28, Aguilera et al., does not explicitly disclose performing the recited steps within a .NET environment. However, it would have been obvious to one of ordinary skill in the art to implement these functions within the .NET framework, given the teachings of the Java API. In addition, Java and .NET are considered to be alternatives to each other, both being class libraries designed to run on a virtual machine. Therefore, it is considered an obvious implementation detail to choose either Java or .NET based on the developer's familiarity.

As per claim 30, the combination of Aguilera et al. as modified, teaches wherein evaluating a node context further comprises: identifying an optimized branch opcode from which the new node will depend (Appendix A, pg 8); identifying one or more other nodes that depend from the optimized branch opcode that include a similar expression as the new node (Appendix A, pg 8); and

if minimum threshold number of the one or more other nodes is not met, modifying the optimized branch opcode to a generic branch opcode that can process the number of one or more other nodes more efficiently than the optimized branch opcode can (Graefe et al.: pg. 361, last paragraph of section 5, "If there is no gain in using a dynamic access module, the decision tree can by an empty function"). Graefe et al. teaches a dynamic optimization technique that can be placed at any point in a query evaluation plan, such as a branch node input to hold the optimization algorithm. The decision to either add or remove an optimization technique is provided by Graefe et al., and would be decided upon at the same time.

As per claim 33, Aguilera et al. as modified does not explicitly teach the modifying further comprises removing the branch node if the branch node references only one other opcode node other than the opcode node to be removed.

However, it would have been obvious to one of ordinary skill in the art at the time of the invention that the modifying further comprises removing the branch node if the branch node references only one other opcode node other than the opcode node to be removed. This is considered obvious because having a branch node where the tree no longer branches is superfluous and would waste unneeded time and space.

As per claim 35, Aguilera et al. as modified teaches the modifying further comprises implementing an optimized processing function in the branch node if the removal of the branch node creates a context in which the optimized processing function would increase efficiency of the branch node processing (Graefe et al.: pg. 361, last paragraph of section 5, "If there is no gain in using a dynamic access module, the decision tree can by an empty function").

As per claim 36, this claim is substantially the same as claim 23 above, and is rejected on the same grounds.

As per claim 40, this claim is substantially the same as claim 21 above, and is rejected on the same grounds.

As per claim 42, this claim is substantially the same as claim 19 above, and is rejected on the same grounds.

As per claim 43, Aguilera et al. as modified, does not explicitly disclose wherein the steps are performed in a Common Language Runtime (CLR) environment. It would have been obvious to one of ordinary skill in the art at the time of the invention that CLR is an execution environment for the .NET library, and is an alternative to the Java Virtual Machine, which Java runs on. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the steps in a Common Language Runtime (CLR) environment, if they were implementing the invention in a .NET framework rather than Java. Choice of implementation language is not considered a patentable distinction.

As per claim 44, Aguilera et al. as modified, discloses wherein for each given (key, value) pair corresponding to one branch, the key is the value of matching literals in the branch, and the value identifies the branch to which the literal belongs (Graefe et al., pg 360). These steps are implicit for making use of a hash table, or other efficient lookup index. See applicant spec. pg. 22, lines 3-7, "the value of the literal is hashed to derive an index entry" (emphasis added), in particular.

As per claim 45, this claim is substantially the same as claim 44 above, and is rejected on the same grounds.

As per claim 46, this claim is substantially the same as claim 44 above, and is rejected on the same grounds.

As per claim 47, this claim is substantially the same as claim 44 above, and is rejected on the same grounds.

As per claim 48, this claim is substantially the same as claim 44 above, and is rejected on the same grounds.

7. Claims 19, 31 and 37 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Yicun Wu whose telephone number is 571-272-4087. The examiner can normally be reached on 8:00 am to 4:30 pm, Monday -Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, MOHAMMAD ALI can be reached on 571-272-4105. The fax phone numbers for the organization where this application or proceeding is assigned are 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 571-272-2100.

Yicun Wu
Patent Examiner
Technology Center 2100

September 30, 2008

/Yicun Wu/

Primary Examiner, Art Unit 2169